1 Introduction

The area of computer graphics is a vast and ever expanding field. In the simplest sense computer graphics are images viewable on a computer screen. Applications extend into such processes as engineering design programs and almost any type of media. The images are generated using computers and likewise, are manipulated by computers. Underlying the representation of the images on the computer screen is the mathematics of Linear Algebra. We will explore the fundamentals of how computers use linear algebra to create these images, and branch off into basic manipulation of these images.

2 2-Dimensional Graphics

Examples of computer graphics are those of which belong to 2 dimensions. Common 2D graphics include text. For example the vertices of the letter H can be represented by the following data matrix $D$:

$$D = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 2 & 0 & 1 & 2
\end{bmatrix}$$

Letter Shown Here:

$$H$$

2.1 Scaling

There are many types of transformations that these graphics can undergo, the first one we will consider is scaling. A point in the $xy$ plane coordinates
are given by \((x,y)\) or \(A=\begin{bmatrix} X \\ Y \end{bmatrix}\).

The scaling transformation is given by the matrix \(S=\begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}\). Where the \(C_i\)s are scalars. The scaling transforms the coordinates \((x,y)\) into \((C_1x, C_2y)\). In mathematical terms the transformation is given by the multiplication of the matrices \(S\) and \(A\):

\[
\begin{bmatrix} S \\ \end{bmatrix} \begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} C_1X \\ C_2Y \end{bmatrix}
\]

**Example:** Now let’s examine how scaling affects our example H. Let \(C_1=2\) and \(C_2=2\), our new coordinates for D are:

\[
D = \begin{bmatrix} 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 2 & 4 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\]

The new H is now twice as long in the X-direction and twice as long in the Y-direction.

Now that we have examined the scaling transformation, let’s move on to the other two basic transformations that underly the movement of a figure on a computer screen, translating and rotating.

### 2.2 Translation

Let’s revisit our example matrix A with coordinates \((X,Y,1)\) or \(\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}\). The post-translational coordinates of A can be obtained by multiplying A by the matrix \(I T\), where I represents the \(I_3\) Identity Matrix, and T represents the column vector containing the translation coordinates for A. Mathematically speaking translation is represented by:

\[
\begin{bmatrix} I \\ T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & X_0 \\ 0 & 1 & Y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} X+X_0 \\ Y+Y_0 \\ 1 \end{bmatrix}
\]

**Example:** Let us now revisit our example letter and H and show how it is affected by an arbitrary translation. Let us set our arbitrary translational vector to \(T=\begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}\), then \(\begin{bmatrix} I \\ T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}\), and the final matrix can be given by:
$$\begin{bmatrix} I & T \end{bmatrix} \begin{bmatrix} D \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 3 & 3 & 3 \\ 3 & 4 & 5 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Our example letter H has now moved 2 units in the positive X-direction and 3 units in the positive Y-direction.

2.3 Rotation

Next, we have the transformation of rotation. A more complex transformation, rotation changes the orientation of the image about some axis, in our case either X or Y. Clockwise rotations have the rotational matrix

$$R(-\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

but all rotational matrices are given in a variation of the form of the rotational matrix for counter-clockwise rotation

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$ The counter-clockwise rotation of matrix A is given by:

$$\begin{bmatrix} R(\theta) \end{bmatrix} \begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} X\cos(\theta) - Y\sin(\theta) \\ X\sin(\theta) + Y\cos(\theta) \\ 1 \end{bmatrix}.$$

**Example:** Now let us visually examine a rotation of $90^\circ$ of our example letter H:

Mathematically speaking the rotation of $90^\circ$ occurs when $\theta = \pi/2$. So now we can view the mathematics that made this transformation possible:

$$\begin{bmatrix} R(\theta) \end{bmatrix} \begin{bmatrix} D \end{bmatrix} = \begin{bmatrix} \cos(\pi/2) & -\sin(\pi/2) & 0 \\ \sin(\pi/2) & \cos(\pi/2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$
If we refer to both the earlier visualization and mathematical display we see that letter H has been rotated 90° about the origin.

### 2.4 Composite Transformations

Now that we have gone over these basic transformations, it is now time to combine them to examine the net result of how multiple transformations affect an image, movement. Movement of the image is caused by the multiplication of the scaling, translational, rotational, and other transformational matrices with the coordinate matrix. The result of these matrix multiplications are called Composite Transformations. To finish up our view in the world of 2-Dimensional graphics, lets mathematically examine a composite transformation of our example letter H using the scaling, translational, and rotational matrices we’ve already established. The composite transformation of H can be represented by:

\[
\begin{bmatrix}
S & T & R(\theta) & D
\end{bmatrix} =
\begin{bmatrix}
0 & -1 & -2 & 0 & -1 & -2 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
2 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 2 \\
0 & 1 & 3 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 20 & 20 & 20 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

### 3 3-Dimensional Graphics

Now that we’ve effectively explored the region of 2-Dimensional, we can begin to explore the region of 3-Dimensional graphics. 3-Dimensional graphics live in \( \mathbb{R}^3 \) versus 2-Dimensional graphics which live in \( \mathbb{R}^2 \). 3-Dimensional graphics have a vast deal more applications in comparison to 2-Dimensional graphics, and are, likewise, more complicated. We will now work with the variable Z, in addition to X and Y, to fully represent coordinates on the X, Y, and Z axes, or simply space.

Homogeneous 3-Dimensional Coordinates For every point \((X,Y,Z,)\) in \( \mathbb{R}^3 \), there exists a point \((X,Y,Z,1)\) on the \( \mathbb{R}^4 \) plane. We call the second pair of coordinates Homogeneous Coordinates. (Note: Homogeneous coordinates
can not be added or multiplied by scalars. In R3 they may only be multi-
plied by 4x4 matrices.) If $H \neq 0$, then:

- $X = \frac{x}{H}$
- $Y = \frac{y}{H}$
- $Z = \frac{z}{H}$

We now call $(x, y, z, H)$ homogeneous coordinates for $(X, Y, Z)$ and $(X, Y, Z, 1)$

**Example:** Find homogeneous coordinates for the point $(4, 2, 7)$

Solution:

Refer to our earlier equations for finding the coordinates $(x, y, z)$. Let’s show two solutions for the question.

Solution 1: Let $H = \frac{1}{2}$:

$x = \frac{4}{\frac{1}{2}} = 8$, $y = \frac{2}{\frac{1}{2}} = 4$, $z = \frac{7}{\frac{1}{2}} = 14$

$(x, y, z) = (8, 4, 14)$

Solution 2: Let $H=2$:

$x = \frac{4}{2} = 2$, $y = \frac{2}{2} = 1$, $z = \frac{7}{2} = 3.5$

$(x, y, z) = (2, 1, 3.5)$

**3.1 Visualization**

When we view 3-Dimensional objects on computer screens, we do not actually see the images themselves, but rather we see projections of these objects onto the viewing plane, or computer screen. The object we see is determined by a set amount of straight line segments, whose endpoints $P_1, P_2, P_3, \ldots, P_n$ are represented by coordinates $(x_1, y_1, z_1)$,
\((x_2, y_2, z_2), (x_3, y_3, z_3), \ldots, (x_n, y_n, z_n)\). These line segments and coordinates are placed in the memory of a computer. The corresponding coordinate matrix can be represented by:

\[
P = \begin{bmatrix}
X_1 & X_2 & \ldots & X_n \\
Y_1 & Y_2 & \ldots & Y_n \\
Z_1 & Z_2 & \ldots & Z_n \\
\end{bmatrix}
\]

### 3.1.1 Perspective Projection

Let’s assume that a 3-Dimensional object displayed on a computer screen is being mapped onto the XY-plane. We say that a perspective projection maps each point \((x, y, z)\) onto an image point \((x^*, y^*, 0)\). We call the point where the image’s coordinates, it’s projected coordinates, and the eye position meet the center of projection.

**Example:** Using homogeneous coordinates represent the perspective projection of the matrix \(P\).

Solution: If we scale the coordinates by \(\frac{1}{1-z/d}\), then \((x, y, z, 1) \rightarrow (x\frac{1}{1-z/d}, y\frac{1}{1-z/d}, 0, 1)\)

Now we can represent \(P\) as:

\[
P = \begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & -1/10 & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix} = \begin{bmatrix}
x \\
y \\
0 \\
1-\frac{1}{10}
\end{bmatrix}
\]

### 3.2 Scaling

In 3-Dimensions, scaling moves the coordinates \((X,Y,Z)\) to new coordinates \((C_1, C_2, C_3)\) where the \(C - i\) are scalars. Scaling in 3-Dimensions is exactly like scaling in 2-Dimensions, except that the scaling occurs along 3 axes, rather than 2. Note that if we view strictly from the XY-plane the scaling in the Z-direction can not be seen, if we view strictly from the XZ-plane the scaling in the Y-direction can not be seen, and if we view strictly from the YZ-plane then the scaling in the X-direction can not be seen. The scaling matrix in 3-Dimension is represented by:
The scaling transformation, like in 2-Dimensions, is represented by the matrix multiplication of the Scaling Matrix and coordinate matrix $A$:

$$S = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix}$$

Example: Give the matrix for a scaled cube with original coordinates given by $A = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 2 & 2 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & -2 & -2 & 0 & 0 & -2 & -2 \end{bmatrix}$, and the scaling matrix is given by $S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$

Solution:

$$S \cdot A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 2 & 2 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & -2 & -2 & 0 & 0 & -2 & -2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 4 & 4 & 0 & 0 & 4 & 4 & 0 \\ 0 & -6 & -6 & 0 & 0 & -6 & -6 \end{bmatrix}$$

From calculating the new matrix we can see that the cube isn’t scaled in the X-direction, scaled to twice it’s size in the positive Y-direction, and scaled to triple it’s size in the negative Z-direction.

### 3.3 Translation

Like scaling, Translation is exactly the same in space as in 2-Dimensions, with the exception of the use of the Z-axis. If we were to view a translating object moving in either the positive or negative Z-direction strictly from the perspective of the XY-plane, then it would appear to us that image is increasing or decreasing, respectively, in size. In reality this is just how the computer establishes the object is translating in space. This also applies for the other directions with respect to the other planes. Translation
in 3-dimensions is represented by the matrix multiplication of the transla-
tion vector \( T = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \), and the coordinate matrix \( A = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \).

Mathematically speaking we can represent the 3-Dimensional translation transformation with:

\[
\begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X + X_0 \\ Y + Y_0 \\ Z + Z_0 \\ 1 \end{bmatrix}
\]

**Example:** Give the matrix for the translation of a point \((5, 3, -8, 1)\) by the vector \( p = (-4, -6, 3) \)

Solution: The matrix that maps \((X, Y, Z, 1) \rightarrow (X - 4, Y - 6, Z + 3, 1)\) is given by

\[
\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 5 \\ 3 \\ -8 \\ 1 \end{bmatrix}, \text{ so:}
\]

\[
\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ -8 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ -5 \\ 1 \end{bmatrix}
\]

### 3.4 Rotation

We finally arrive at rotation in 3-Dimensions. Just like scaling and translating, rotation in three dimensions is fundamentally the same as rotation in 3-Dimensions, with the exception that we also can rotate about the Z-axis. The rotations about the X, Y, and Z axes are given by:

- \( R(\theta)_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cos(\theta) & -Sin(\theta) \\ 0 & Sin(\theta) & Cos(\theta) \end{bmatrix} \)

(Rotates the y-axis towards the z-axis)
\[ R(\theta)_Y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \]

(Rotates the z-axis towards the x-axis)

\[ R(\theta)_Z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

(Rotates the x-axis towards the y-axis)

We can display the post-reflection coordinates as:

\[ R(\theta)_X: (X,Y,Z) \rightarrow (X,Y\cos(\theta) - Z\sin(\theta), Z\cos(\theta) + Y\sin(\theta)) \]

\[ R(\theta)_Y: (X,Y,Z) \rightarrow (X\cos(\theta) + Z\sin(\theta), Y, -X\sin(\theta) + Z\cos(\theta)) \]

\[ R(\theta)_Z: (X,Y,Z) \rightarrow (X\cos(\theta) - Y\sin(\theta), Y\cos(\theta) + X\sin(\theta), Z) \]

**Example:** Give the final Rotational Matrix, \( R \), for an image that is first rotated about the X-axis 47°, then about the Y-axis 52°, and then about the Z-axis -30°.

Solution: \( R \) can be mathematically represented by the matrix multiplications of the 3 rotational matrices of the X, Y, and Z axes. Mathematically speaking:

\[
\begin{align*}
R &= \begin{bmatrix} R(\theta)_X & R(\theta)_Y & R(\theta)_Z \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(47°) & -\sin(47°) \\ 0 & \sin(47°) & \cos(47°) \end{bmatrix} \begin{bmatrix} \cos(52°) & 0 & \sin(52°) \\ 0 & 1 & 0 \\ -\sin(52°) & 0 & \cos(52°) \end{bmatrix} \begin{bmatrix} \cos(-30°) & -\sin(-30°) & 0 \\ \sin(-30°) & \cos(-30°) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} .533 & .308 & .788 \\ .158 & .879 & -.450 \\ -.831 & .365 & .420 \end{bmatrix}
\end{align*}
\]
4 Conclusion

Today we have only scratched the surface of the world of computer graphics. The advancement of computer graphics has made technology more functional and user-friendly. Mainstream applications of computer graphics can be seen in every type of media including animation, movies, and video games. The breadth of computer graphics range from basic pixel art and vector graphics to the incredibly realistic animation of high end 3D gaming programs, such as games typically played on Microsoft©Xbox 360. Applications of these graphics also extend into the world of science. Currently biologists are using computer graphics for molecular modeling. Through these crucial visualizations scientists are able to make advances in drug and cancer research, that otherwise would not be possible. A major future application of computer graphics lies in the domain of virtual reality. In virtual reality one is able to experience a synthesized computer environment just as if it were a natural environment. It would appear that "the sky is the limit" for the world of computer graphics, but at the basis of it all lies the mathematics of linear algebra.

5 References


